

불가능하다는 것을 아는 것

이재현

2022.09.15

Abstract

프로그램의 동작을 자동으로, 정확하게, 유한 시간 내에 검증하는 분석기를 만드는 것은
아쉽게도 불가능하다. 하지만, 다행히도 불가능하다는 사실 자체를 알기 때문에 한계와
타협하며 최선의 분석기를 만들 수 있다.

무언가를 이룰 수 없다는 사실을 깨닫는 것은 일반적으로 절망스러운 일로 여겨진다. 초등학교 3학년이 되던 해, 산타 할아버지가 없다는 것을 알게 되었다. 그리고 그로 인해 더 이상 산타 할아버지에게 크리스마스 선물을 받을 수 없음을 알았을 때 크게 실망했다. 영생을 추구한 사람들도 있다. 중국을 통일한 진시황은 죽음을 두려워하여 불로초를 찾으려 했다. 아이러니하게도 진시황은 불로불사를 위해 복용한 수은에 중독되어 생을 마감했다. 개발자들은 자신이 만드는 프로그램이 자신의 의도대로 동작하기를 희망한다. 그렇지만, 안타깝게도 프로그램의 동작을 자동으로, 정확하게, 유한 시간 내에 검증하는 것은 불가능하다.

무한히 많은 프로그램 동작을 완벽히 검증할 수는 없다. 튜링의 정지 문제가 한 예시이다. 우리가 가진 프로그래밍 언어 체계로는 어떤 프로그램이 결국에는 멈출지, 아니면 무한히 돌아갈지 유한 시간 내에 판단하는 알고리즘을 구현할 수 없다. 라이스는 이에 더해서, 자명하지 않은 프로그램의 의미 성질(semantic property)을 검증하는 알고리즘은 존재할 수 없음을 증명했다. 컴퓨터와 프로그램을 가지고 무엇이든 해결할 수 있을 것만 같은데, 풀 수 없는 문제가 있다는 것을 실망스럽게 느껴진다.

그렇지만, 다른 관점에서 보면 무언가를 할 수 없다는 사실을 아는 것 자체가 행운이기도 하다. 소크라테스의 일화가 이를 뒷받침한다. 소크라테스의 제자가 델포이 신전에서 신탁을 받았는데, 신탁의 내용은 "소크라테스는 모든 사람들 중 가장 현명하다."는 것이었다. 소크라테스는 그 신탁이 정말인지를 알아보기 위해 자신보다 더 현명한 사람을 찾아보려 했다. 그는 지혜롭다고 여겨지는 사람들을 만났지만, 실은 그들이 모르는 것까지도 아는 척 하고 있음을 알게 된다. 그리고 "어쨌든 나는 적어도 이 사람들보다는 바로 이 점에서 조금은 더 지혜로운 것 같다. 나는 내가 알지 못하는 것들을 알지 못한다고 생각도 한다는 점에서 말이다."라는 결론을 내린다. 그렇기에 어떤 것이 불가능하다는 것을 알게 된 순간, 우리는 이전보다 더 현명해진 것이라 여길 수 있다.

무언가를 이룰 수 없다는 것을 인정한 순간부터, 우리는 타협을 시작할 수 있다. 프로그램을 자동으로, 정확하게, 유한 시간 내에 검증하는 분석기를 만들 수 없다는 사실을 인정하면, 조건을 하나씩 포기하는 방법으로 타협을 시도할 수 있다. 우리가 보편적으로 사용하는 타입 체계는 가끔 올바른 프로그램을 틀렸다고 분류하기도 한다. 하지만, 타입 체계가 있으면 프로그래밍 중에 실수를 빠르게 잡아낼 수 있다. 자동으로 테스트들을 생성해 실행시키며 에러를 찾는 퍼즈 테스트 기법도 있다. 결국 에러가 존재하는지를 열거식으로 찾는 방식이기에 시간 제한을 두어야 하지만, 보안 취약점을 찾는 용도로 사용된다. 완벽한 해답은 없지만, 프로그래밍 언어 체계가 허용하는 한계 안에서 최선을 다할 수는 있다.

불가능한 일을 마주했을 때, 그 사실에 절망할 수도 있지만, 반대로 내 삶의 범위가 더 넓어졌다고도 받아들일 수 있다. 그리고 그것을 인정하면서 가능과 불가능의 경계에서 새로운 해답을 타협할 수 있다. 우리는 인간이 필멸의 존재라는 것을 알고 있다. 그렇기에 우리는 유한한 시간을 소중히 사용한다. 마찬가지로 우리는 선물을 줄 산타 할아버지가 없다는 것을 알고 있지만, 휴일 자체가 선물이라는 것도 알고 있다. 그렇기에 우리는 올해도 가족과 함께하는 크리스마스를 기다린다.